

ALIBABA CLOUD

阿里云

IP地理位置库
最佳实践

文档版本：20210903

 阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.从零开始接入阿里云IP地理位置库	05
2.离线版数据服务更新	14

1.从零开始接入阿里云IP地理位置库

目前该产品在做内部调整，不支持新购、续费、升级等操作，订单有效期内可继续正常使用，本文档仅供订单期内用户参考。请留意服务到期时间，做好备选方案，感谢您的支持理解。

产品对接指南

1、API对接常用参数（后文产品对接中会用到）

获取AccessKeyID、Secret

获取实例ID（即InstanceId）

regionId目前仅支持cn-hangzhou节点，国外用户强烈不推荐使用在线版

2、API快速入门

3、API在线调试

产品使用指南



注意

本部分主要介绍代码如何使用及演示，如果已经看过产品文档，已理解如何使用，可忽略以下文档，或直接查看[离线库升级](#)文档。

1、IP地理位置库（在线版）对接指南

在线版服务支持用户调用阿里云API获取IP地理定位能力，在具体对接时，我们建议您通过集成阿里云开放平台SDK来调用服务（[前往下载SDK](#)）。

下文中以集成java版本阿里云SDK为例，演示如何对接在线版IP地理位置库服务：

- （1）、使用idea开发工具创建Maven全新项目
- （2）、添加jar包依赖，下载jar文件坐标
- （3）、创建com.example包下的DescribeIpv4Online.java文件，填写AK信息（即前文提到的AccessKeyID、Secret）

```
package com.example;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;

import java.util.*;

import com.aliyuncs.geoip.model.v20200101.*;

public class Describepv4Online {

    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile(
            "cn-hangzhou", //此处配置regionId（固定是cn-hangzhou）
            "AccessKeyID",
            "Secret"
        );
        IAcsClient client = new DefaultAcsClient(profile);

        Describepv4LocationRequest request = new Describepv4LocationRequest();
        request.setIp("221.206.131.10");
        try {
            Describepv4LocationResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

(4)、运行Describepv4Online.java示例，在控制台得到以下结果

```
{
  "requestId": "8BEE7785-0755-420A-9EF7-1F3BCE5F53F6",
  "ip": "221.206.131.10",
  "country": "中国",
  "province": "黑龙江省",
  "city": "牡丹江市",
  "county": "阳明区",
  "isp": "中国联通",
  "countryCode": "CN",
  "countryEn": "China",
  "provinceEn": "Heilongjiang",
  "cityEn": "Mudanjiang",
  "longitude": "129.634645",
  "latitude": "44.596328"
}
```

此时返回值是小驼峰格式，如果想返回标准API大驼峰格式可以使用以下代码代替，或者直接使用 `response.getCity()` 等方法返回具体值，不用考虑字段大小写。

```
HttpResponse httpResponse = client.doAction(request);
System.out.println("responseURL:\n" + new String(httpResponse.getHttpContent()));
```

2、IP地理位置库（[离线版](#)）对接指南

注意

为了不发生混乱我们重新创建个Maven项目用于离线版的演示

离线版服务支持用户将加密IP库离线下载到本地，在本地搭建服务使用（[前往下载离线SDK](#)）。

购买开通服务后可以在控制台 [下载文件/API调用](#)（详情参考离线升级文档）得到以下三个文件

.lic为授权文件

.dex为离线库文件

离线SDK文件（选择需要的开发语言版本即可，下文以java版本为例演示对接）

例如：

 v1.20200819111516-version_standard-20210220.lic
 v1.20200824190642.jar
 v1.20200902131803.dex

下文基于以上3个文件（SDK以java版本做演示），开始演示如何使用。

集成sdk方式：

（1）、复制jar文件到项目中，文件位置可自行设置，并添加引用

File > Project Structure > Modules > Dependencies > + > JARs or directories

添加commons-lang3、guava、commons-codec、fastjson、slf4j-api的Maven坐标并下载

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>untitled1</artifactId>
  <version>1.0-SNAPSHOT</version>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.5.RELEASE</version>
  </parent>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <repositories>
    <repository>
      <id>sonatype-nexus-staging</id>
      <name>Sonatype Nexus Staging</name>
      <url>https://oss.sonatype.org/service/local/staging/deploy/maven2</url>
      <releases>
        <enabled>true</enabled>
      </releases>
      <snapshots>
        <enabled>true</enabled>
      </snapshots>
    </repository>
  </repositories>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-lang3</artifactId>
      <version>3.10</version>
    </dependency>
    <dependency>
      <groupId>com.google.guava</groupId>
      <artifactId>guava</artifactId>
      <version>28.2-jre</version>
    </dependency>
    <dependency>
      <groupId>commons-codec</groupId>
      <artifactId>commons-codec</artifactId>
      <version>1.11</version>
    </dependency>
  </dependencies>
</project>
```

```
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.68</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.7.26</version>
</dependency>
</dependencies>
</project>
```

(2)、创建com.example.DescribIpv4Offline.java，配置授权文件和离线库文件访问路径

```
package com.example;

import com.alibaba.sec.client.FastIPGeoClient;
import com.alibaba.sec.domain.FastGeoConf;
import com.alibaba.sec.exception.FastIPGeoException;
import com.alibaba.sec.license.exception.LicenseException;
import org.springframework.stereotype.Service;

@Service
public class DescribIpv4Offline {

    private static final String DATA_FILE_PATH = "v1.20201013183640.dex";
    private static final String LICENSE_FILE_PATH = "v1.20201013183640.lic";

    public static void main(String[] args) throws Exception {
        FastGeoConf geoConf = new FastGeoConf();
        geoConf.setDataFilePath(DATA_FILE_PATH);
        geoConf.setLicenseFilePath(LICENSE_FILE_PATH);

        //普通对象方式
        //FastIPGeoClient fastIpGeoClient = new FastIPGeoClient(geoConf);

        //单例模式
        FastIPGeoClient fastIpGeoClient = FastIPGeoClient.getSingleton(geoConf);
        String result = null;
        try {
            result = fastIpGeoClient.search("221.206.131.10");
            System.out.println(result);
        } catch (LicenseException | FastIPGeoException e) {
            e.printStackTrace();
        }
    }
}
```

运行DescribIpv4Offline.java示例，在控制台得到以下结果

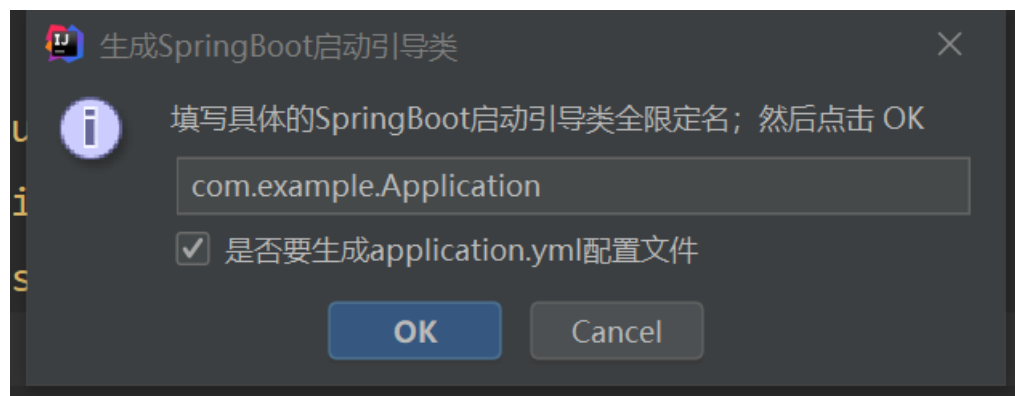
```
{
  "country": "中国",
  "city": "牡丹江市",
  "county_code": "231003",
  "isp": "中国联通",
  "province_en": "Heilongjiang",
  "latitude": "44.596328",
  "county": "阳明区",
  "country_en": "China",
  "city_code": "231000",
  "province_code": "230000",
  "city_en": "Mudanjiang",
  "country_code": "CN",
  "routes": "中国联通",
  "province": "黑龙江省",
  "isp_code": "100026",
  "longitude": "129.634645"
}
```

SpringBoot 方式：

注意

SpringBoot 方式仅仅是为了演示方便使用，具体业务要根据情况而定，http/rpc方式对性能以及qps有很大损失

使用idea的JBLSpringBootApplication插件创建启动引导类



(1)、创建com.example.service.GeoipService，配置授权文件和离线库文件访问路径

```
package com.example.service;

import com.alibaba.sec.client.FastIPGeoClient;
import com.alibaba.sec.domain.FastGeoConf;
import com.alibaba.sec.exception.FastIPGeoException;
import com.alibaba.sec.license.exception.LicenseException;
import org.springframework.stereotype.Service;
import javax.annotation.PostConstruct;

@Service
public class GeoipService {

    private static final String DATA_FILE_PATH = "v1.20201013183640.dex";
    private static final String LICENSE_FILE_PATH = "v1.20201013183640.lic";
    private FastIPGeoClient fastIpGeoClient ;

    @PostConstruct
    public void init() throws Exception {
        FastGeoConf geoConf = new FastGeoConf();
        geoConf.setDataFilePath(DATA_FILE_PATH);
        geoConf.setLicenseFilePath(LICENSE_FILE_PATH);
        //普通对象
        //fastIpGeoClient = new FastIPGeoClient(geoConf);
        //单例模式
        fastIpGeoClient = FastIPGeoClient.getSingleton(geoConf);
    }

    public String searchIpv4(String ip) throws Exception {
        String result = null;
        try {
            result = fastIpGeoClient .search(ip);
        } catch (LicenseException | FastIPGeoException e) {
            e.printStackTrace();
        }
        return result;
    }
}
```

(2)、创建com.example.controller.GeoipController.java

```
package com.example.controller;

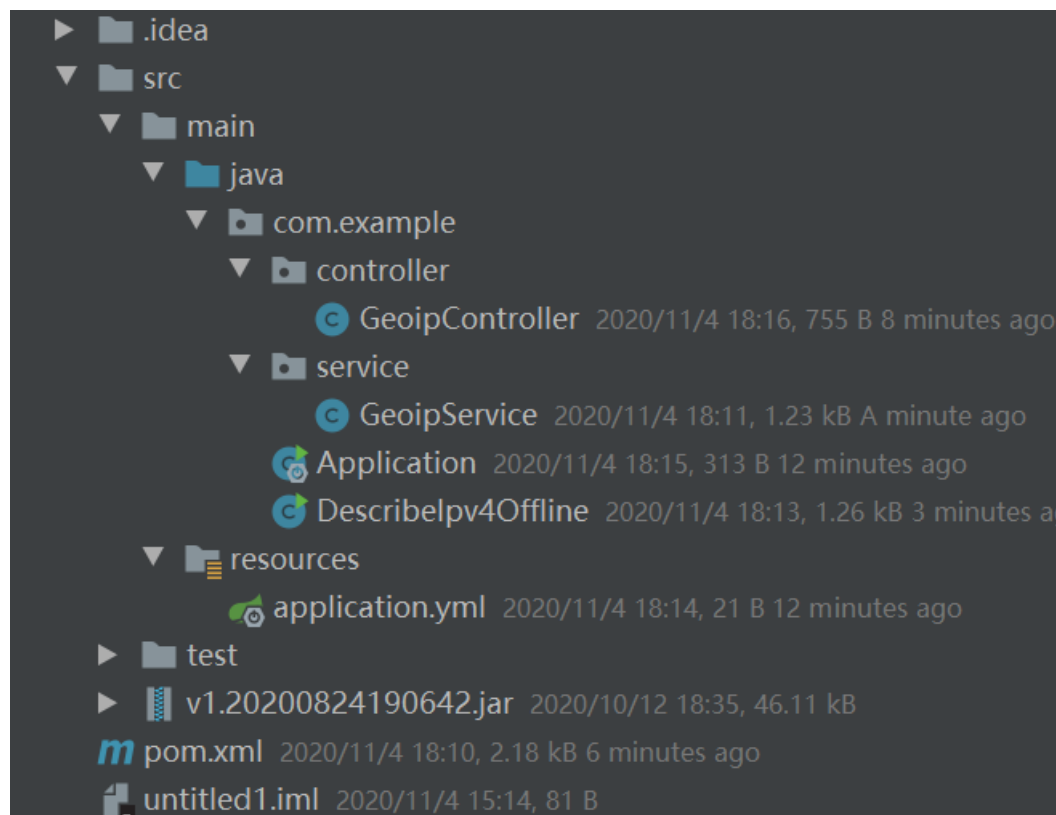
import com.example.service.GeoipService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RequestMapping("/http")
public class GeoipController {

    @Autowired
    private GeoipService geoipService;

    @GetMapping("/offline/ipv4/{ip}")
    public String offline(@PathVariable String ip) throws Exception {
        return geoipService.searchIpv4(ip);
    }
}
```

离线版项目结构参考



(3)、启动SpringBoot项目，在浏览器访问

<http://127.0.0.1:8080/http/offline/ipv4/221.206.131.10>

JSON

```
country : 中国
city : 牡丹江市
county_code : 231003
isp : 中国联通
province_en : Heilongjiang
latitude : 44.596328
county : 阳明区
country_en : China
city_code : 231000
province_code : 230000
city_en : Mudanjiang
country_code : CN
routes : 中国联通
province : 黑龙江省
( isp_code : 100026
longitude : 129.634645
```

2. 离线版数据服务更新

目前该产品在做内部调整，不支持新购、续费、升级等操作，订单有效期内可继续正常使用，本文档仅供订单期内用户参考。请留意服务到期时间，做好备选方案，感谢您的支持理解。

升级指南

离线版因为可以脱离公网使用，仅有查询功能，所以离线库升级无法在离线sdk中继续完成，可通过API的方式获取离线库文件下载地址，通过与本地文件版本号对比选择合适时间下载并替换完成升级。

注意

如果对数据精确度要求较高推荐经常更新离线库文件，阿里数据源一般24小时更新一次。授权文件通常因为购买的服务发生变配升级、续费而产生更新，强烈推荐发现新版授权文件后立即更新，确保服务升级续费在您本地生效。sdk更新频率较低，如果发现有升级也推荐更新（sdk更新可能要重启服务，建议在业务低峰期处理）。

1、手动升级

到[阿里云控制台](#)页面直接下载最新的离线库文件、sdk、授权文件配置离线文件路径重新启动服务或者替换实例进行升级

优点：方便、简单、快速使用服务

缺点：当前节点可能需要停止服务

2、自动升级（推荐）

通过[API](#)获取离线库文件、sdk、授权文件等下载地址，与本地版本号对比，自动下载文件、替换文件、更换实例或重启服务进行升级

优点：可[升级集群](#)，保持文件最新，可不停止服务进行升级

缺点：升级方案需要自行设置

以下三个API用于升级离线产品，演示代码默认使用SpringBoot（仅仅是为了方便显示json返回值）

（1）、通过[DescribeGeoipInstances](#)获取所有已购买的产品实例列表（instanceId即实例ID，也可[手动获取](#)）返回结果如下所示：

```
{
  "productCode": "geoipv4offline",
  "instanceId": "geoipv4offline_cn-*",
  "versionCode": "version_standard",
  "status": "OPEN",
  "createTime": "2020-08-19T03:15Z",
  "createTimestamp": 1597806917000,
  "expireTime": "2022-02-22T16:00Z",
  "expireTimestamp": 1645545600000,
  "maxQps": 150000,
  "maxQpd": null
}
```

（2）、通过 [DescribeGeoipInstanceDataInfos](#) 根据传入参数获取离线版实例数据信息，需要使用 [instanceId](#) 参数。

在[从零开始接入阿里云IP地理位置库](#)在线版项目com.example.service.GeoipService.java添加如下代码

```
//根据传入参数获取离线版实例数据信息
public DescribeGeoipInstanceDataInfosResponse DescribeGeoipInstanceDataInfos(String instanceId) {
    DescribeGeoipInstanceDataInfosRequest request = new DescribeGeoipInstanceDataInfosRequest();
    request.setInstanceId(instanceId);

    DescribeGeoipInstanceDataInfosResponse response = null;
    try {
        response = client.getAcsResponse(request);
        System.out.println(new Gson().toJson(response));
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        System.out.println("ErrCode:" + e.getErrCode());
        System.out.println("ErrMsg:" + e.getErrMsg());
        System.out.println("RequestId:" + e.getRequestId());
    }

    return response;
}
```

在[从零开始接入阿里云IP地理位置库](#)在线版项目com.example.controller.GeoipController.java添加如下代码

```
@GetMapping("/online/dataInfos/{instanceId}")
public DescribeGeoipInstanceDataInfosResponse dataInfos(@PathVariable String instanceId) {
    return geoipService.DescribeGeoipInstanceDataInfos(instanceId);
}
```

在浏览器访问

http://127.0.0.1:8080/http/online/dataInfos/geoipv4offline_cn-*

参数为购买服务的实例ID

```
{
  "requestId": "BB2DB17C-EFA9-4ABF-8D57-02C02156CD0E",
  "dataInfos": [
    {
      "type": "IPV4_DATA",
      "version": "1.20210420122927",
      "updateTime": "2021-04-20T04:45Z",
      "updateTimestamp": 1618893902000,
      "downloadCount": 0
    },
    {
      "type": "LICENSE",
      "version": "1.20210222123044",
      "updateTime": "2021-02-22T04:30Z",
      "updateTimestamp": 1613968246000,
      "downloadCount": 202
    },
    {
      "type": "JAVA_SDK",
      "version": "1.20210113140642",
      "updateTime": "2021-01-13T08:30Z",
      "updateTimestamp": 1610526600000,
      "downloadCount": 96
    },
    {
      "type": "CPP_SDK",
      "version": "1.20200908161000",
      "updateTime": "2020-09-08T08:15Z",
      "updateTimestamp": 1599552901000,
      "downloadCount": 86
    },
    {
      "type": "PYTHON_SDK",
      "version": "1.20200903171000",
      "updateTime": "2020-09-03T09:19Z",
      "updateTimestamp": 1599124765000,
      "downloadCount": 85
    }
  ]
}
```

API返回了具体的sdk、授权文件、离线库等类型，根据类型继续调用API即可得到具体的文件下载地址

(3)、通过 [DescribeGeoipInstanceDataUrl](#) 根据传入参数获取离线版实例数据下载链接，需要使用 `instanceId`、`dataType`（上个接口type字段）参数。

在[从零开始接入阿里云IP地理位置库](#)在线版项目com.example.service.GeoipService.java添加如下代码

```
//根据传入参数获取离线版实例数据下载链接
public DescribeGeoipInstanceDataUrlResponse DescribeGeoipInstanceDataUrl(String instanceId, String dataType) {
    DescribeGeoipInstanceDataUrlRequest request = new DescribeGeoipInstanceDataUrlRequest();
    request.setInstanceId(instanceId);
    request.setDataType(dataType);

    DescribeGeoipInstanceDataUrlResponse response = null;
    try {
        response = client.getAcsResponse(request);
        //可以用 Getter方法拿到下载地址url或直接打印json
        //System.out.println(response.getFixedDomainDownloadUrl());
        System.out.println(new GsonBuilder().disableHtmlEscaping().create().toJson(response));
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        System.out.println("ErrCode:" + e.getErrCode());
        System.out.println("ErrMsg:" + e.getErrMsg());
        System.out.println("RequestId:" + e.getRequestId());
    }

    return response;
}
```

在[从零开始接入阿里云IP地理位置库](#)在线版项目com.example.controller.GeoipController.java添加如下代码

```
@GetMapping("/online/dataUrl/{instanceId}/{dataType}")
public DescribeGeoipInstanceDataUrlResponse dataUrl(@PathVariable String instanceId, @PathVariable String dataType) {
    System.out.println(instanceId);
    System.out.println(dataType);
    return geoipService.DescribeGeoipInstanceDataUrl(instanceId, dataType);
}
```

在浏览器访问

http://127.0.0.1:8080/http/online/dataUrl/geoipv4offline_cn-*/IPV4_DATA

结果如下所示：得到当前类型的下载地址，可通过代码下载到本地替换数据完成离线库更新，其中fixedDomainDownloadUrl返回值为固定域名，推荐优先使用此地址下载，若需要配置[安全策略](#)，可将file.geoip.alidns.com域名加白名单。

```
{
  "fixedDomainDownloadUrl": "https://file.geoip.alidns.com/product/v1/ipv4/common/v1.20210420082401.dex?Expires=*",
  "requestId": "65E15C91-7FC4-4B99-9C94-692E9DEF9F2",
  "downloadUrl": "http://geoip-sdk-user.oss-cn-zhangjiakou.aliyuncs.com/product/v1/ipv4/common/v1.20210420082401.dex?Expires=*",
}
```

http://127.0.0.1:8080/http/online/dataUrl/geoipv4offline_cn-*/LICENSE

http://127.0.0.1:8080/http/online/dataUrl/geoipv4offline_cn-*/JAVA_SDK

`http://127.0.0.1:8080/http/online/dataUrl/geoipv4offline_cn-*/CPP_SDK`

`http://127.0.0.1:8080/http/online/dataUrl/geoipv4offline_cn-*/PYTHON_SDK`

注意

获取的链接地址具有时效性，超时后需要重新请求获得地址

升级说明

离线库文件升级后要重新启动实例或平滑重启服务，新的实例才会生效（数据在第一次实例化后就已经全部加载到内存中，后续直接从内存读取），以java为例：

注意

如果下载离线文件或生成实例时有异常或错误，请捕获异常，此时不要替换或重启服务，否则影响原程序运行。

一、通过构造方法初始化后的实例重启

```
FastIPGeoClient fastIpGeoClient = new FastIPGeoClient(geoConf);
```

在自行下载升级离线数据后，需要配置新的实例，如：

```
FastIPGeoClient fastIpGeoClient2 = new FastIPGeoClient(geoConf2);
```

在自定义想替换的时机重新赋值，如：

```
fastIpGeoClient = fastIpGeoClient2;
```

二、通过静态方法（单例模式）初始化后的实例重启

因为是将加载字段、license文件路径、数据文件路径封装成一个实例，获取的实例不会重复创建。

```
FastIPGeoClient fastIpGeoClient = FastIPGeoClient.getSingleton(geoConf);
```

1、可以采用非单例方法来包装单例，然后在自定义想替换的时机重新实例化对象即可。

2、更改license或数据文件的路径，重新加载单例模式的实例。